

How to send a jar file from a Java ME app using Bluetooth

This article shows how to send a jar file from a Java ME application using Bluetooth. The code will prompt the user to turn Bluetooth on if it is not active.



This article needs to be updated: If you found this article useful, please fix the problems below then delete the `{{ArticleNeedsUpdate}}` template from the article to remove this warning.

Reasons: hamishwillee (31 Aug 2012)

The attached zip file with source is corrupted. Needs to be recreated.

Ensure Bluetooth is on and get address

This first snippet is initiated when a command or some user action is detected for sending the jar file. It gets the Bluetooth address and if this fails because Bluetooth is off raises a system wide alert to request the user to turn it on (only in Series 40 mobiles.)

```
new Thread() {
    public void run() {
        try {
            LocalDevice ld=LocalDevice.getLocalDevice();
            String frndlyName=ld.getFriendlyName();
            String btAddr=ld.getBluetoothAddress();
            //Call a class or method for searching the devices in range and displaying them.
        } catch(Exception e) {
            Alert a=new Alert("Error", "Your Bluetooth is not ON or set to Hidden mode! Please
change it and click 'Send via Bluetooth' again", null, AlertType.ERROR);
            a.setTimeout(Alert.FOREVER);
            display.setCurrent(a, null);
        }
    }
}.start();
```

Initiate device search

The next part of the code initiates the device search method of the *DiscoveryListener* interface.

```
LocalDevice devLocal = LocalDevice.getLocalDevice();
DiscoveryAgent discoverAgent = devLocal.getDiscoveryAgent();
discoverAgent.startInquiry(DiscoveryAgent.GIAC, deviceDiscover);''
```

Display devices in range

The third action is to display the devices in range (discovered in the previous section). The *DiscoveryListener* interface will handle the device search. The devices got from that should have to be handled for displaying in a List or any UI component.

Note this is not shown here.

Sending the file

The final code snippet below shows how to handle the file properties and send the file to the selected Client.

```
Connection connection = Connector.open(btConnectionURL);
ClientSession cs = (ClientSession)connection;
HeaderSet hs = cs.createHeaderSet();
```

```
cs.connect(hs);
//Set the File Name at Client
hs.setHeader(HeaderSet.NAME, val);
//Set the File's extension at Client
hs.setHeader(HeaderSet.TYPE, extension);
//Set the File's Size at Client
hs.setHeader(HeaderSet.LENGTH, new Long(file.length));
//Push the file to Client
Operation putOperation=cs.put(hs);
OutputStream outputStream = putOperation.openOutputStream();
outputStream.write(bout.toByteArray());
//File push complete
outputStream.close();
putOperation.close();
cs.disconnect(null);
connection.close();''
```

Example application

- [Media:Sharing a Jar file via Bluetooth from a J2ME Appln.zip](#)